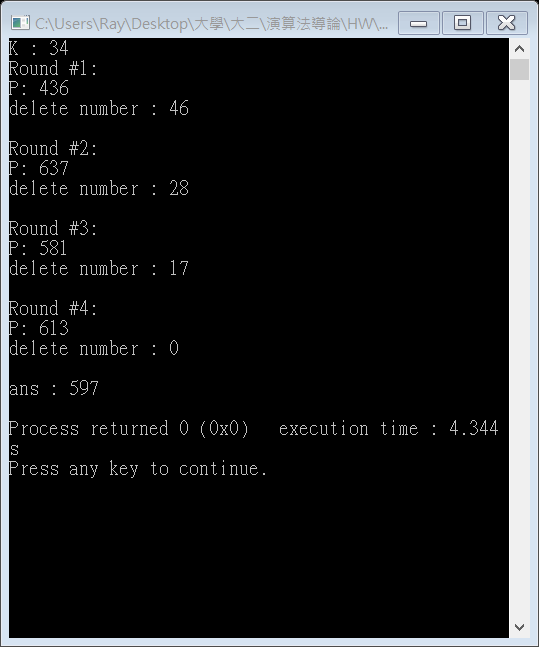
**程式說明 :**

以課程prune and search的方法，完成一個selection problem程式，隨機方式產生100個1至1000的正整數。接著從鍵盤輸入K，每回合，selection problem演算法會先找出P值，並刪減部分的數字。

最終會找到第K大的數字為何。

**程式執行畫面 :** 

**原始程式碼 :**

#include<bits/stdc++.h>

#include<time.h>

using namespace std;

int main()

{

int k,now=0,t=0,ans=0,de=100;

vector<int> num,test;

srand(time(NULL));

for(int i=0;i<100;i++)

{

num.push\_back(rand()%1000);

}

cout<<"K : ";

cin>>k;

while(1)

{

t++;

cout<<"Round #"<<t<<": "<<endl;

int B\_num=0,S\_num=0,L\_num=0;

vector<vector<int> > pru;

vector<int> bigger,same,little,temp,mid;

for(int i=0,j=1;i<num.size();i++,j++)

{

if(j==6)

{

pru.push\_back(temp);

temp.clear();

j=1;

}

temp.push\_back(num[i]);

}

if(temp.size()<5)

{

int need=5-temp.size();

for(int i=0;i<need;i++)

{

temp.push\_back(10000);

}

}

pru.push\_back(temp);

temp.clear();

for(int i=0;i<pru.size();i++)

{

sort(pru[i].begin(),pru[i].end());

mid.push\_back(pru[i][2]);

}

sort(mid.begin(),mid.end());

int mm=mid.size()/2,x=1;

int m=mid[mm];

while(m==10000)

{

m=mid[mm-x];

x++;

}

cout<<"P: "<<m<<endl;

for(int i=0;i<num.size();i++)

{

if(num[i]>m)

{

bigger.push\_back(num[i]);

B\_num++;

}

else if(num[i]==m)

{

same.push\_back(num[i]);

S\_num++;

}

else

{

little.push\_back(num[i]);

L\_num++;

}

}

if((B\_num+now)<k)

{

bigger.clear();

now+=B\_num;

if((now+S\_num)<k)

{

same.clear();

now+=S\_num;

if((now+L\_num)>=k)

{

if(L\_num<=5)

{

sort(little.begin(),little.end());

reverse(little.begin(),little.end());

ans=little[k-now-1];

break;

}

}

num.clear();

num.assign(little.begin(),little.end());

}

else if((now+S\_num)>=k)

{

if(S\_num<=5)

{

sort(same.begin(),same.end());

reverse(same.begin(),same.end());

ans=same[k-now-1];

break;

}

now+=S\_num;

num.clear();

num.assign(same.begin(),same.end());

}

}

else if((B\_num+now)>=k)

{

if(B\_num<=5)

{

sort(bigger.begin(),bigger.end());

reverse(bigger.begin(),bigger.end());

ans=bigger[k-now-1];

break;

}

num.clear();

num.assign(bigger.begin(),bigger.end());

}

bigger.clear();

same.clear();

little.clear();

mid.clear();

pru.clear();

cout<<"delete number : "<<de-num.size()<<endl<<endl;

de=num.size();

}

cout<<"delete number : "<<de-num.size()<<endl<<endl;

de=num.size();

cout<<"ans : "<<ans<<endl;

return 0;

}

**簡要心得 :**

運用課堂學到的prune and search來做，因為是第一次寫，所以也花了不少的時間，倒是沒遇上甚麼大問題，原本寫完之後還在苦惱要怎麼是測資，想好幾天才想到，另外用一個sort排好直接拿第幾大的數就好了，這個應該是我這次作業裡想最久也最憨的問題了。